REMARKS

Status of the Claims

Claims 1 through 26 are pending. Claims 1 through 26 stand rejected under 35 U.S.C. § 102(e). Applicants propose amending claims 1, 2, 4, 5, 7, 10, 11, 20, and 26. No new matter has been added. Applicants propose canceling claims 3, 6, 8, 16, and 23.

Rejections Under 35 U.S.C. § 102(e)

Claims 1 – 26 stand rejected under 35 U.S.C. § 102(e) as allegedly being anticipated by Shrivastava et al., (U.S. Pub. 2004/0243576 A1), (hereinafter referred to as "Shrivastava"). For the reasons set forth below, the Applicants traverse the rejection and respectfully request reconsideration.

Applicants have disclosed systems and methods for notifying computing applications that changes in a database have taken place that affect the data that the computing applications retrieve from the database. By way of background, the application explains as follows:

> **[0050]** For every query that is submitted to the database server with a subscription request, the query processor maintains a mechanism that enables it to detect whether a change in the underlying data will affect the result of this query. Such mechanism may comprise a notification manager (as described above). For example, given an email client application that subscribes with a multitude of independent queries like the following ones:
>
> SELECT name, subject
> FROM users, mail
> WHERE users.id = mail.recipient AND user.name = 'Joe'
>
> and
>
> SELECT name, subject
> FROM users, mail
> WHERE users.id = mail.recipient AND user.name = 'Jack'
>
> **[0051]** The notification manager (NM) removes the parameters from the query and stores them in a parameter table of the form as follows:

> CREATE TABLE parameter_table (param_1 NVARCHAR(20))
>
> **[0052]** In the example provided there is one parameter table per query template. The number of columns in the table as well as the type depend on number and type of the parameters in the original query. In this example, the name of a user was given the type NVARCHAR(20). Per subscription, one row with the actual parameter(s) is inserted in the parameter table.
>
> . . .
>
> **[0056]** Using query templates, the change detection query can be formulated as,
>
> SELECT name, subject
> FROM
> (SELECT name, subject
> FROM users, mail_delta
> WHERE users.id = mail.recipient) as delta
> JOIN
> parameter_table
> ON delta.name = parameter_table.param_1
>
> The plan is independent of the number of subscriptions – their individual parameters are stored in parameter_table and are addressed by the join predicate. (Application, ¶¶ 0050 – 0056).

Consistent with this description, amended claim 1 recites a method for providing notifications of changes in a database system comprising:

> 1. A method for providing notifications of changes in a database system, comprising:
> **receiving a plurality of query statements for querying a database system, each query statement corresponding to a computing application that has subscribed to receive notification of changes in the database system affecting data retrieved from the database system by the computing application;**
> creating a subscription template **from the plurality of query statements;**

> generating a parameter table **from the plurality of query statements, the parameter table comprising for each query statement a constant representing a query value and a subscription statement identification value uniquely identifying a subscription associated with the particular query statement;**
>
> **in response to a change in the data in the database,** performing a join between said parameter table and said subscription template to generate a query;
>
> **executing the query on the database system to identify query statements in the plurality of query statements affected by the change in the data in the database; and**
>
> **communicating notification to a computing application corresponding to an identified query statement, said notification indicating a change in the data in the database has occurred.**

In order for a reference to anticipate claim 1, it must teach the *entirety* of the recited method. The undersigned respectfully submits that Shrivastava does not teach at least the emphasized claim language and cannot possibly teach or even suggest the recited method.

Paragraphs 76 through 86 of Shrivastava, which are referenced by the Office in support of its rejection, disclose a system and method for automatically generating a query statement to search for particular objects or entries in a directory information tree ("DIT") that is stored within relational tables. (*See* Shrivastava, ¶ 76.) Shrivastava teaches using templates to convert an arbitrary LDAP search filter into a single SQL statement where a base template provides the basic framework for generating an SQL statement. (*Id.* at ¶ 78.) Additional templates are used to fill in specific portions of the base template. The LDAP is converted into a single SQL statement based upon the base template. (*Id.* at ¶ 79.)

Thus, Shrivastava is directed to creating a SQL statement corresponding to an LDAP search filter. In contrast to claim 1, Shrivastava does not disclose a method "for providing notifications of changes in a database system." Indeed, providing notifications of changes in a database system is not even a consideration of the system disclosed in Shrivastava. Accordingly, Shrivastava does not disclose or suggest the body of the recited claim language.

For example, Shrivastava does not teach or suggest "receiving a plurality of query statements for querying a database system, **each query statement corresponding to a computing application that has subscribed to receive notification of changes in the**

database system affecting data retrieved from the database system by the computing application." In fact, the portions of Shrivastava referenced in the Office Action do not appear to discuss "subscrib[ing] to receive notifications of changes in [a] database system" at all. The referenced portions of Shrivastava certainly do not disclose or suggest "each query statement corresponding to a computing application that has subscribed to receive notification of changes in the database system affecting data retrieved from the database system by the computing application."

Furthermore, Shrivastava does not teach "generating a parameter table **from the plurality of query statements, the parameter table comprising for each query statement a constant representing a query value and a subscription identification value uniquely identifying a subscription associated with the particular query statement.**" The Office analogizes that the "catalog tables" mentioned by Shrivastava correspond to the recited parameter table. (Office Action dated 6/13/07, p. 3). But the catalog tables disclosed by Shrivastava do not comprise "for each query statement a constant representing a query value and a subscription identification value uniquely identifying a subscription associated with the particular query statement."

Still further, Shrivastava does not teach "**in response to a change in the data in the database**, performing a join between said parameter table and said subscription template to generate a query." The Office alleges that Shrivastava teaches performing a join between a parameter table and a parameterized subscription template. (Office Action dated 6/13/07, p. 3). For all of the reasons set out in the Reply filed August 13, 2007, Applicant respectfully disagrees with the Office's analysis. Moreover, claim 1 has been amended to recite that the join is performed **"in response to a change in the data in the database."** Shrivastava simply does not disclose performing the recited join "in response to a change in the data in the database." Rather, the join disclosed by Shrivastava that the Office cites to (Shrivastava, ¶¶ 0076-0086) as allegedly teaching this element is made as part of a process for generating a query corresponding to a LDAP search filter and not in response to changes in the data of a database.

Finally, Shrivastava does not disclose or suggest "**executing the query on the database system to identify query statements in the plurality of query statements affected by the change in the data in the database,**" and "**communicating notification to**

a computing application corresponding to an identified query statements, said notification indicating the change in the data in the database has occurred." As noted above, providing notifications of changes in a database system does not appear to be a consideration of the system disclosed in Shrivastava. Accordingly, Shrivastava does not disclose or suggest "executing the query on the database system to identify query statements . . . affected by the change in the data" and "communicating notification to a computing applications . . . indicating the change in a change in the data in the database has occurred."

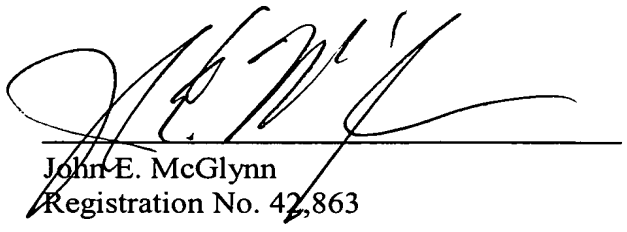
Therefore, because it does not teach all of the recited language, Shrivastava does not anticipate claim 1 and its dependent claims. Independent claims 10, 11, 20, and 26, and all claims depending therefrom are likewise not anticipated by Shrivastava for analogous reasons. Withdrawal of the rejections under 35 U.S.C. § 102(e) is respectfully solicited.

CONCLUSION

The undersigned respectfully submits that pending claims are allowable and the application in condition for allowance. A Notice of Allowance is respectfully solicited.

Examiner Thai is invited to call the undersigned in the event a telephone interview will advance prosecution of this application.

Date: December 11, 2007



John E. McGlynn
Registration No. 42,863

Woodcock Washburn LLP
Cira Centre
2929 Arch Street, 12th Floor
Philadelphia, PA 19104-2891
Telephone: (215) 568-3100
Facsimile: (215) 568-3439